Readme File – CS21B052

Classes and their functionalities-

- **User Class**
  i) This class stores all the necessary details of the user.
  ii) All the personal information is stored in private fields to hide user-sensitive data.
  iii) The class has getters and setters of private fields as methods.
  iv) The class has the method “checkValid” which checks whether the given string contains all integers or not.

- **Admin Class**
  i) This class has all the information and functionalities related to Admin.
  ii) One need to enter the User Id and Password to sign in as Admin. The user Id and password are set in code as-
  User Id- Admin
  Password- admin@123
  iii) The admin has two functionalities-
    a) To add money daily in the ATM.
    b) To view the available balance in ATM.
  iv) The class has the method “addMoney” which stores the number of notes of all denominations in an array and returns it to the ATM’s balance.
  v) The method for showing available balance is written in “MoneyTransaction” class.

- **Account Class**
  i) This class contains all the functionalities which are “only” related to the User and not the ATM.
  ii) The class has a vector of “User” objects which stores the information of all the users dynamically.
  iii) The class has following methods- “signUp”, “signIn”, “getIdx”, “resetPin”, “resetPhoneNo”, “resetEmail”.

a) signUp method-
   - Method to create account for new users.
   - It checks if the account number entered by user is a 5-digit number or not.
   - Further, it checks if the entered account number already exists in the database or not.
   - If the account number is valid, the user should enter the PIN which should be a number. Then the user should re-enter the PIN and it should match with the PIN.
   - Then, the user should enter the mobile number and e-mail address to create the account. The mobile number should be a 10- digit number.

b) signIn method-
   - Method for login of existing users.
   - To login, the user should enter account number and PIN.
   - If both account number and PIN are valid, the account is signed in.
   - If the login credentials are correct, the index where the details are stored in the vector is taken for further functionalities.
   - The user can enter wrong PIN for a maximum of three times.

c) resetPin method-
   - When user enters wrong password for 3 times, they get an option to reset the PIN.
   - The user should enter the correct mobile number to reset the PIN.
   - The user should re-enter the new PIN to set it successfully.

d) resetPhoneNo method-
   - The user has an option to reset the mobile number.
   - They should enter the correct old mobile number to set the new one.

e) resetEmail method-
    - The user has an option to reset the email address too.
    - They should enter the correct old email address to set the new one.

- **CashDispenser (Abstract) Class**
  i) The class contains two arrays- one array “cash” contains the number of notes of each denomination and another array “denomination” contains the denomination of respective index in “cash” array.

- **Functionality Interface**
  i) This interface contains all the functionalities required to proceed the transaction.
  ii) The functionalities include depositing money, checking cash availability, getting index of required denomination.

- **MoneyTransaction Class**
  i) This class contains all the methods related to transactions between bank account and ATM.
  ii) The class has following methods- getDenomination, cashAvailability, depositMoney, withdrawMoney, showAvailableBalance.
  a) getDenomination method-
      - Gives the denomination of provided index from the “denomination” array.
  b) cashAvailability method-
      - It is a method to check if the required amount is available in ATM or not.
      - The availability of cash is checked from higher denominations to lower denominations if the required number of notes/coins are available in the ATM or not.

c) depositMoney method-
    - A method to let user deposit money in ATM.
    - The users select the denomination they want to deposit and enter -1 when they have deposited all the money.
    - After that, the updated account balance is shown.
    - The number of notes deposited by user are stored denomination wise in an array to make changes in "cash" array.
d) depositMoney method with array as parameter-
    - Polymorphism is used here for depositMoney method.
    - This method updates the "cash" array with the money deposited by the user.
e) withdrawMoney method-
    - A method to let user withdraw the money from ATM.
    - If the "cashAvailability" method returns true, then the cash is withdrawn from the ATM and "cash" array is simultaneously updated.
    - The cash is withdrawn in the same way as it checks in "cashAvailability" method.
f) showAvailableBalance method-
    - A method for admin to show available balance in ATM.
    - The method shows the number of each denomination available in ATM.

Use of OOPs concept in ATM-

- **Use of Inheritance-**
    - Inheritance is used in "MoneyTransaction" class. The class uses the fields from the class "CashDispenser", where the information related to number of notes available in ATM is stored.
    - Inheritance has helped to create new class to specify new implementation using the data from the super class.

- **Use of Polymorphism-**
    - Polymorphism is used in “depositMoney” method in “MoneyTransaction” class.
    - One method without any parameter collects the data about the number of notes deposited by the user for respective denominations and stores it in an array.
    - Another method with that array as parameter updates the data in “cash” array.
    - Polymorphism has helped to increase readability as we can use same name for the methods having same functionalities.
- **Use of Interface-**
    - Interface is used to keep a track of all the functionalities used in the transaction.
    - The interface “Functionality” contains those methods which are implemented in “MoneyTransaction” class.

- **Use of Encapsulation-**
    - Objects are created for all classes.
    - Each classes have different functionalities. All the functionalities are encapsulated into the classes.